



Appl. No. 09/662,258  
Amdt. dated: May 4, 2004  
Reply to Office Action of December 4, 2003

Docket No. SUN-P4175  
(811173-000094)

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings of claims in the application:

**RECEIVED**

LISTING OF CLAIMS

**MAY 11 2004**

- 
1. (Cancelled) Technology Center 2100
2. (Cancelled)
3. (Currently Amended) A method for representing an application programming interface (API) definition for an object-oriented library, said method comprising:  
creating a public list including all public classes and interfaces defined in said object-oriented library, said public list including a class sublist for each of said public classes, each said class sublist including all direct and indirect public superclasses of a class and excluding private classes; and  
storing said list.
4. (Previously Presented) The method of claim 3 wherein said object-oriented library is a Java™ package.
5. (Cancelled)
6. (Cancelled)

7. (Cancelled)

8. (Cancelled)

9. (Cancelled)

10. (Currently Amended) A program storage device readable by a machine, embodying  
a program of instructions executable by the machine to perform a method to  
represent an application programming interface (API) definition for an object-  
oriented library, the method comprising:  
creating a public list including all public classes and interfaces defined in said object-  
oriented library, said public list including a class sublist for each of said public  
classes, each said class sublist including all direct and indirect public  
superclasses of a class and excluding private classes; and  
storing said list.

11. (Previously Presented) The program storage device of claim 10 wherein said object-  
oriented library is a Java™ package.

12. (Cancelled)

13. (Cancelled)

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Currently Amended) An apparatus for representing an application programming interface (API) definition for an object-oriented library, said apparatus comprising:  
means for creating a public list including all public classes and interfaces defined in  
said object-oriented library, said public list including a class sublist for each of  
said public classes, each said class sublist including all direct and indirect public  
superclasses of a class and excluding private classes; and  
means for storing said list.

18. (Previously Presented) The apparatus of claim 17 wherein said object-oriented library is a Java™ package.

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (New) A method for representing an application programming interface (API) definition for an object-oriented library, said method comprising:  
step for creating a public list including all public classes and interfaces defined in said object-oriented library, said public list including a class sublist for each of said public classes, each said class sublist including all direct and indirect public superclasses of a class and excluding private classes; and  
step for storing said list.
23. (New) The method of claim 22 wherein said object-oriented library is a Java<sup>TM</sup> package.
24. (New) A method for determining a program hierarchy, said method comprising:  
receiving an application programming interface (API) definition file for an object-oriented library, said API definition file including a list of public elements in said object-oriented library, each element comprising a class or interface, each of said public elements including a sublist of all public hierarchically-related elements that are a parent of the element and excluding private classes; and  
indicating a first public element is a direct parent of a second public element when said first public element is represented in the sublist for said second public element and said first public element is not represented in the sublist for any other public element listed in the sublist for said second public element.

25. (New) The method of claim 24 wherein said object-oriented library is a Java<sup>TM</sup> package.
26. (New) The method of claim 24, further comprising comparing a first program hierarchy reconstructed from a first API definition file with a second program hierarchy reconstructed from a second API definition file; and indicating an error when said first program hierarchy is inconsistent with said second program hierarchy.
27. (New) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to determine a program hierarchy, said method comprising: receiving an application programming interface (API) definition file for an object-oriented library, said API definition file including a list of public elements in said object-oriented library, each element comprising a class or interface, each of said public elements including a sublist of all public hierarchically-related elements that are a parent of the element and excluding private classes; and indicating a first public element is a direct parent of a second public element when said first public element is represented in the sublist for said second public element and said first public element is not represented in the sublist for any other public element listed in the sublist for said second public element.

28. (New) The program storage device of claim 27 wherein said object-oriented library is a Java<sup>TM</sup> package.

29. (New) The program storage device of claim 27, said method further comprising:  
comparing a first program hierarchy reconstructed from a first API definition file  
with a second program hierarchy reconstructed from a second API definition  
file; and

indicating an error when said first program hierarchy is inconsistent with said second  
program hierarchy.

30. (New) An apparatus for determining a program hierarchy, said apparatus  
comprising:  
means for receiving an application programming interface (API) definition file for an  
object-oriented library, said API definition file including a list of public  
elements in said object-oriented library, each element comprising a class or  
interface, each of said public elements including a sublist of all public  
hierarchically-related elements that are a parent of the element and excluding  
private classes; and

means for indicating a first public element is a direct parent of a second public  
element when said first public element is represented in the sublist for said  
second public element and said first public element is not represented in the  
sublist for any other public element listed in the sublist for said second public  
element.

31. (New) The apparatus of claim 30 wherein said object-oriented library is a Java™ package.

32. (New) The apparatus of claim 30, said apparatus further configured to:  
compare a first program hierarchy reconstructed from a first API definition file with  
a second program hierarchy reconstructed from a second API definition file; and  
indicate an error when said first program hierarchy is inconsistent with said second  
program hierarchy.

33. (New) A method for determining a program hierarchy, said method comprising:  
step for receiving an application programming interface (API) definition file for an  
object-oriented library, said API definition file including a list of public  
elements in said object-oriented library, each element comprising a class or  
interface, each of said public elements including a sublist of all public  
hierarchically-related elements that are a parent of the element and excluding  
private classes; and  
step for indicating a first public element is a direct parent of a second public element  
when said first public element is represented in the sublist for said second public  
element and said first public element is not represented in the sublist for any  
other public element listed in the sublist for said second public element.

34. (New) The method of claim 33 wherein said object-oriented library is a Java<sup>TM</sup> package.
35. (New) The method of claim 34, further comprising
- C1 step for comparing a first program hierarchy reconstructed from a first API definition file with a second program hierarchy reconstructed from a second API definition file; and
- step for indicating an error when said first program hierarchy is inconsistent with said second program hierarchy.